

# Making our HTML Look Nicer

## Cascading Style Sheets

### Chapter 2

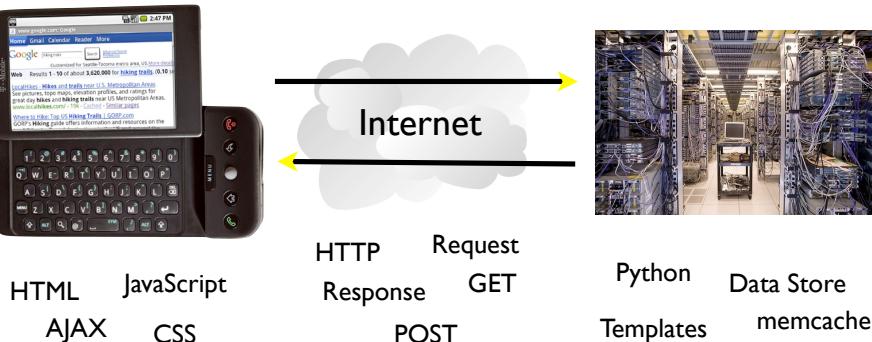
Supporting Material for "Using Google App Engine" - O'Reilly and Associates  
[www.appspotlearn.com](http://www.appspotlearn.com)

open.michigan

Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.  
<http://creativecommons.org/licenses/by/3.0/>.

Copyright 2009, Charles Severance

UNIVERSITY OF MICHIGAN



## CSS Already?

- Back in the ancient days (i.e. before 2007) we got used to thinking that CSS was “advanced HTML” and that it was “difficult”
- With modern browsers, we should never bother with the old ways to change font, color, alignment, etc etc.
- CSS - Just Do It!

From Simple to Advanced CSS with no changes to the HTML

## The big picture...



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<title>University of Michigan</title>
...
```

In the modern era of web design we represent content and meaning in HTML and layout in CSS.

```
@import "/CSS/graphical.css" ***!
p{text strong ,verbose ,verbose p ,verbose h2{text-indent:-876em;position:absolute}
p{text strong a{text-decoration:none}
p{text em{font-weight:bold;font-style:normal}
div.alert{background:#eee;border:1px solid red;padding:5em;margin:0 25%}
a img{border:none}
.hot br ,quick br ,dl.feature2 img{display:none}
div#main label,legend{font-weight:bold}}
```

## Separation of concerns / specialization

**Developer**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<title>University of Michigan</title>
...
```

**Designer**

```
@import "/CSS/graphical.css" ***!
p{text strong ,verbose ,verbose p ,verbose h2{text-indent:-876em;position:absolute}
p{text strong a{text-decoration:none}
p{text em{font-weight:bold;font-style:normal}
div.alert{background:#eee;border:1px solid
```

Everyone needs to know some HTML and some CSS and some programming - but to be truly skilled at a professional level requires deep understanding and specialization.

1995

2007

HTML has evolved a \*lot\* over the years - as computers and networks have gotten faster.

## CSS Syntax

- CSS Syntax is very different than HTML
- CSS is a set of “rules” which include a “selector” and one or more “properties” and “values” as well as some punctuation...

```
body {  
  font-family: arial, sans-serif;  
}
```

## Anatomy of a CSS Rule

selector - which part of the document does this rule apply to

```
body {  
  font-family: arial, sans-serif;  
  font-size: 100%;  
}
```

property - which aspect of CSS are we changing

value - What are we setting the property to.

## Partial List of CSS Properties

color	text-decoration
background-color	border-width
visibility (visible/hidden)	border-style
font-family (arial, sans-serif)	border-color
font-size	margin
font-style (italic, normal)	border
font-weight (bold, normal)	padding
text-align	float (left, right, none)
vertical-align	left / top
text-transform (lowercase, etc)	position (static, relative, absolute)
	z-index

<http://lesliefranke.com/files/reference/csscheatsheet.html>

## Applying CSS to our HTML

- Inline - right on an HTML tag - using the style= attribute
- An embedded style sheet in the <head> of the document
- As external style sheet in a separate file

```
<p>Google AppEngine is scalable.</p>
<p style="color: red; font-weight: bold;">
Don't forget it is free (for small sites).</p>
<p style="font-family: sans-serif;
background-color: green">
Google AppEngine is easy to learn.</p>
```

## Inline Styles

We add a `style=` attribute to a tag. We can have one or more properties and values. The styling applies for the scope of the tag.

Google AppEngine is scalable.

**Don't forget it is free (for small sites).**

**Google AppEngine is easy to learn.**

```
<head>
<title>Learning the Google App Engine</title>
<style type="text/css">
  body {
    font-family: arial, sans-serif;
  }
</style>
</head>
<body>
  <h1><a href="index.htm">
    AppEngineLearn</a></h1>
  ....
</body>
```

The selector in the CSS rule indicates where in the displayed content is to be “styled” according to the properties.



## External Style Sheet

- Preferred because it allows easy switching of style across many files.
- On larger teams, it also allows separation of concerns between the developer (HTML) and designer (CSS)

```
<head>
<title>Learning the Google App Engine</title>
<link type="text/css" rel="stylesheet" href="glike.css">
</head>
<body>
  ....
```

```
csev $ ls -l
total 32
-rw-r--r-- 1 csev staff 44 Dec 19 06:06 glike.css
-rw-r--r-- 1 csev staff 679 Dec 19 06:07 index.htm
-rw-r--r-- 1 csev staff 883 Dec 19 05:59 sites.htm
-rw-r--r-- 1 csev staff 679 Dec 19 05:59 topics.htm
csev $
```



## AppEngineLearn

• Sites  
• Topics

### Google App Engine: About

Welcome to the site dedicated to learning the Google Application Engine. We hope you find [www.appspotenginelearn.com](http://www.appspotenginelearn.com) useful.

First: Just work with the tags and fix fonts / colors



## AppEngineLearn

• Sites  
• Topics

### Google App Engine: About

Welcome to the site dedicated to learning the Google Application Engine. We hope you find [www.appspotenginelearn.com](http://www.appspotenginelearn.com) useful.

```
<head>
  <title>Learning the Google App Engine</title>
  <link type="text/css" rel="stylesheet" href="glike.css">
</head>
```

```
<body>
```

```
....
```

```
body {
  font-family: arial, sans-serif;
}
a {
  color: blue;
}
h1 a {
  text-decoration: none;
  color: black;
}
```



## AppEngineLearn

• Sites  
• Topics

### Google AppEngine: About

Welcome to the site dedicated to learning the Google Application Engine. We hope you find [www.appspotenginelearn.com](http://www.appspotenginelearn.com) useful.

```
<head>
  <title>Learning the Google App Engine</title>
  <link type="text/css" rel="stylesheet" href="glike.css">
</head>
```

```
<body>
```

```
  <h1><a href="index.htm">
    AppEngineLearn</a></h1>
```

```
  <ul>
```

```
    <li><a href="sites.htm">Sites</a></li>
    <li><a href="topics.htm">Topics</a></li>
  </ul>
```

```
  <h1>Google App Engine:About</h1>
```



## AppEngineLearn

• Sites  
• Topics

### Google AppEngine: About

Welcome to the site dedicated to learning the Google Application Engine. We hope you find [www.appspotenginelearn.com](http://www.appspotenginelearn.com) useful.

```
a {
  color: blue;
}
```

Select all "a" tags in the document.

```
<head>
  <title>Learning the Google App Engine</title>
  <link type="text/css" rel="stylesheet" href="glike.css">
</head>
```

```
<body>
```

```
  <h1><a href="index.htm">
    AppEngineLearn</a></h1>
```

```
  <ul>
```

```
    <li><a href="sites.htm">Sites</a></li>
    <li><a href="topics.htm">Topics</a></li>
  </ul>
```

```
  <h1>Google App Engine:About</h1>
```



## AppEngineLearn

• Sites  
• Topics

### Google AppEngine: About

Welcome to the site dedicated to learning the Google Application Engine. We hope you find [www.appspotenginelearn.com](http://www.appspotenginelearn.com) useful.

```
h1 a {
  text-decoration: none;
  color: black;
}
```

Select tags where "a" is inside "h1".

## Marking Text for Styling

- Sometimes we cannot use an existing tag in the document for styling so we use the span tag
- Sometimes we want to apply styling to \*some\* but all of the “p” tags using the class= attribute
- Sometimes we want to break the document into “chunks” using the id= attribute

## The “un-tag” <span>

```
<p>Google AppEngine is scalable  
and <span style="color: red">don't  
forget</span> that it is free  
for small sites and is easy to learn.</p>
```

Span does *not* break word  
wrapping - it simply allows us to  
apply styling to a segment of text  
within a paragraph.

Google AppEngine is scalable  
and **don't forget** that it is free for  
small sites and is easy to learn.

## Making Some Tags Special

- We can use the class= to mark tags and then apply styling specifically to those tags.
- We style these sections by writing a rule which starts with a period followed by the name of the class

```
<head>  
<style type="text/css">  
  .abc {  
    color: blue;  
  }  
</style>  
</head>  
<body>  
<p class="abc">First paragraph</p>  
<p>Second Paragraph</p>  
<p class="abc">Third paragraph</p>  
</body>
```

## Making Some Tags Special

First paragraph  
Second Paragraph  
Third paragraph

```
<head>  
<style type="text/css">  
  .abc {  
    color: blue;  
  }  
</style>  
</head>  
<body>  
<p class="abc">First paragraph</p>  
<p>Second Paragraph</p>  
<p class="abc">Third paragraph</p>  
</body>
```

# Defining and Marking Blocks

- To mark out the scope of blocks, we use the `id=` attribute on a tag.
- We also have a new block tag called `<div>` that we often couple with the `id=` attribute on the div or other tag
- While the `class=` attribute can be (re)used over and over on many tags in a document, a particular `id=` value can only be used once in a document.
- `id=` is used to mark a *particular* block in the document

```
<div id="header">
  <h1><a href="index.htm">AppEngineLearn</a></h1>
  <ul>
    <li><a href="sites.htm">Sites</a></li>
    <li><a href="topics.htm" >Topics</a></li>
  </ul>
</div>
```

To apply styling rules to a block of text identified using `id=` we use a `#` character before the `id` value.

```
#header {
  background-color: #dde;
  border-top: 3px solid #36c;
}
```

```
<body>
  <div id="header">
    <h1><a href="index.htm">AppEngineLearn</a></h1>
    <ul>
      <li><a href="sites.htm">Sites</a></li>
      <li><a href="topics.htm" >Topics</a></li>
    </ul>
  </div>
  <div id="content">
    <h2>Google App Engine:About</h2>
    <p>
      Welcome to the site dedicated to
      learning the Google Application Engine.
      We hope you find www.appspotenginelearn.com useful.
    </p>
  </div>
</body>
```

```
<body>
  <div id="header">
    <h1><a href="index.htm">AppEngineLearn</a></h1>
    <ul>
      <li><a href="sites.htm">Sites</a></li>
      <li><a href="topics.htm" >Topics</a></li>
    </ul>
  </div>
  <div id="content">
    <h2>Google App Engine:About</h2>
    <p>
      Welcome to the site dedicated to
      learning the Google Application Engine.
      We hope you find www.appspotenginelearn.com useful.
    </p>
  </div>
</body>
```

```
body {
  font-family: arial, sans-serif;
}
#header {
  background-color: #dde;
  border-top: 3px solid #36c;
}
a {
  color: blue;
}
#header h1 a {
  text-decoration: none;
  color: black;
}
```

```

body {
    font-family: arial, sans-serif;
}
#header {
    background-color: #dde;
    border-top: 3px solid #36c;
}
a {
    color: blue;
}
#header h1 a {
    text-decoration: none;
    color: black;
}

```

## Don't Overuse Span/Div

- Don't add unnecessary div tags when you already have a block tag to add a class to
- Don't invent your own spans when something exists that will work

```

<div class="important">
<p>
    This is
    <span class="heavy">a</span>
    paragraph.
</p>
</div>

```

```

<p class="important">
    This is <strong>a</strong>
    paragraph.
</p>

```

## Color Names

- W3C has listed 16 color names that will validate with an HTML validator.
- The color names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

[http://www.w3schools.com/html/html\\_colors.asp](http://www.w3schools.com/html/html_colors.asp)

**Color Values**  
 HTML colors can be defined as a hexadecimal notation for the combination of Red, Green, and Blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00) and the highest value is 255 (hex #FF). The table below shows the result of combining Red, Green, and Blue light sources:

Color	Color HEX	Color RGB
Black	#000000	rgb(0,0,0)
Red	#FF0000	rgb(255,0,0)
Green	#00FF00	rgb(0,255,0)
Blue	#0000FF	rgb(0,0,255)
Cyan	#00FFFF	rgb(0,255,255)
Magenta	#FF00FF	rgb(255,0,255)
Yellow	#FFFF00	rgb(255,255,0)
White	#FFFFFF	rgb(255,255,255)
Grey	#CCCCCC	rgb(192,192,192)
LightGrey	#D3D3D3	rgb(211,211,211)

**W3C Standard Color Names**  
 W3C has listed 16 color names that will validate with an HTML validator. The color names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

## Colors...

#e2edff

Three Numbers, Red, Green , and Blue - each from 00 - FF (Hexidecimal)

Color Name	Color HEX	Color
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFF4D	
Azure	#F0FFF0	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBBC	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	

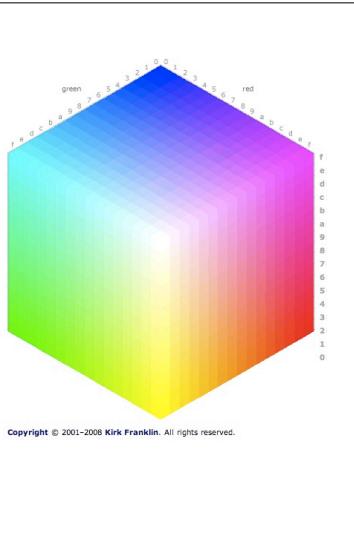
[http://www.w3schools.com/css/css\\_colornames.asp](http://www.w3schools.com/css/css_colornames.asp)

Web-safe colors

## Web-Smart Colors

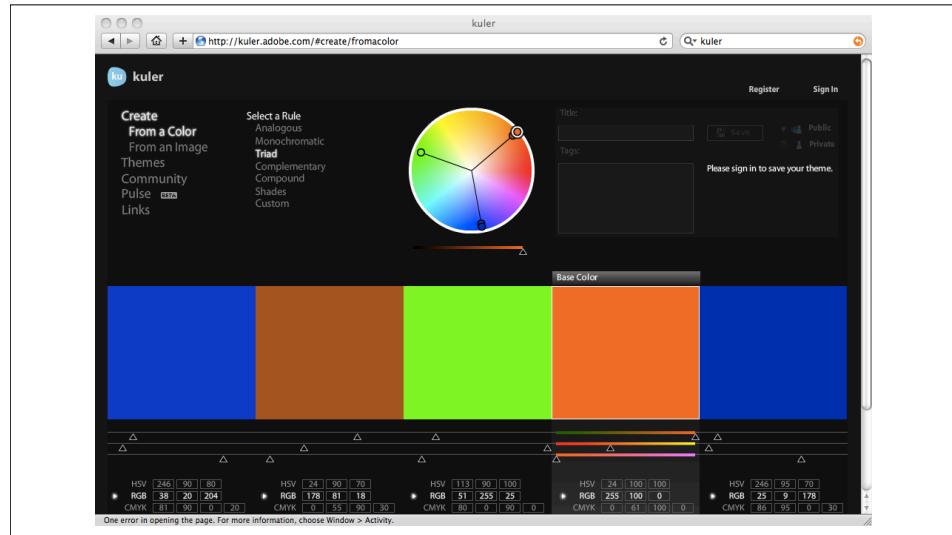
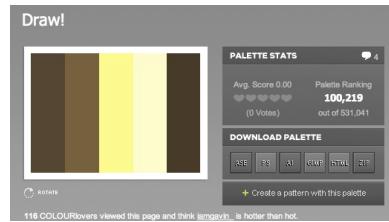
- Web-smart colors are 4096 colors which should easily display accurately on all systems with > 256 colors
- Web smart colors simply use any combination of 00, 11, 22, ... cc, dd, ee, ff
- #0055ee

<http://www.morecrayons.com/palettes/webSmart>  
<http://www.weborum.com/color/wheel.html>



<http://www.colourlovers.com/>

- Picking a combination of complimentary colors is also important - not all colors look good together
- This is a very artistic thing :)



## Fonts

### Header One

Paragraph about ugly default fonts.

- Default fonts are ugly and they have Serifs - which make them harder to read on a screen
- So the first thing I usually want to do is override the font in my document
- And I want to do this everywhere.

Figure 3.5. Highlighting the serifs of a serif font (Georgia)

serif

# Fonts

```
Most Favourite
```

body {  
font-family: "Trebuchet MS", Helvetica, Arial, sans-serif;  
font-size: x-large;  
}

Least Favourite

Fallback fonts: serif, sans-serif, monospace, cursive and fantasy.

# Font Factors

font-size:  
xx-small  
x-small  
small  
medium  
large  
x-large  
xx-large  
14px

font-weight: bold or normal

font-style: normal or italic

text-decoration: none, underline, overline, or  
line-through

# Styling for Links



Post-Click:

Hello there my name is Chuck.

Go ahead and click on [here](#).

Downright Ugly!

# Styling Links

```
a {  
font-weight: bold;  
}  
a:link {  
color: black;  
}  
a:visited {  
color: gray;  
}  
a:hover {  
text-decoration: none;  
color: white;  
background-color: navy;  
}  
a:active {  
color: aqua;  
background-color: navy;  
}
```

link - before a visit  
visited - after it has been visited  
hover - when your mouse is over it  
but you have not clicked  
active - you have clicked it and you  
have not yet seen the new page

Browser default styling for links is  
downright ugly!

**CSS CHEAT SHEET**

SYNTAX	
background	selector {property: value;}
border	border: width color style;
border-left	border-left: width color style;
border-top	border-top: width color style;
font	font: font-family font-size font-style font-weight;
list-style	list-style-type: type;
margin	margin: length;
padding	padding: length;

**Comments**  
/\* Comment \*/

GENERAL	
Class	String preceded by a period
ID	String preceded by a hash mark
selector	Full structure or block of text
span	Inline formation
color	Foreground color
cursor	Appearance of the cursor
display	block, inline, list-item, none
overflow	How content overflowing its box is handled
visibility	visible, hidden, scroll, auto

**Pseudo Selectors**

FONT	
font-style	italic, normal
font-variant	normal, small-caps
font-weight	bold, normal, lighter, bolder, integer (100-900)
font-size	Size of the font
font-family	Specific font(s) to be used

**Media Types**

TEXT	
letter-spacing	Space between letters
line-height	Vertical distance between baselines

**Units**

BOX MODEL	
height	height: width; margin-top; margin-right; margin-bottom; margin-left; padding-top; padding-right; padding-bottom; padding-left;
width	width of the element
border-style	dashed, dotted, double, groove, inset, outset, ridge, solid;
border-width	Width of the border
border-color	Color of the border

**POSITION**

BORDER	
clear	Any floating elements around the element?
float	Floating to a specified side
left	The left position of an element
top	The top position of an element
position	static, relative, absolute
z-index	Element above or below overlapping elements?

**BACKGROUND**

background-color	
background-image	Background image

<http://www.lesliefranke.com/files/reference/csscheatsheet.html>

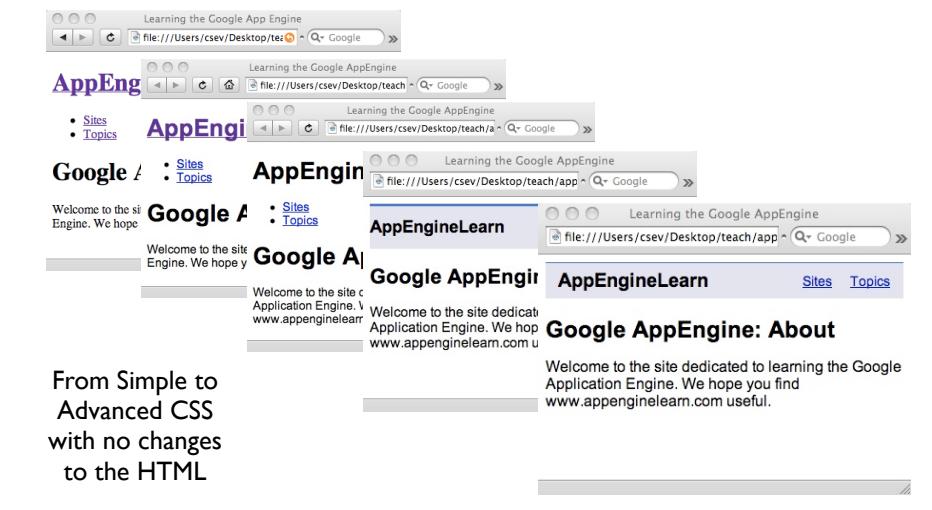
# Web Developer Tools

- You pretty much need to use Firefox for serious website development
- Important plugins:
  - Web Developer - Chris Pedrick
  - FireBug - Joe Hewitt

<http://addons.mozilla.org/>

Using the Firefox Web Developer Plugin you can see the blocks which make up your page.

# Changing Page Layout in CSS



From Simple to  
Advanced CSS  
with no changes  
to the HTML

## Two kinds of elements

- Inline - affects how text looks
  - strong, span
- Block - Containers that can be laid out
  - Paragraphs, etc
- CSS can change a tag from inline to block

```
#navigation li {  
    display: inline;  
}
```

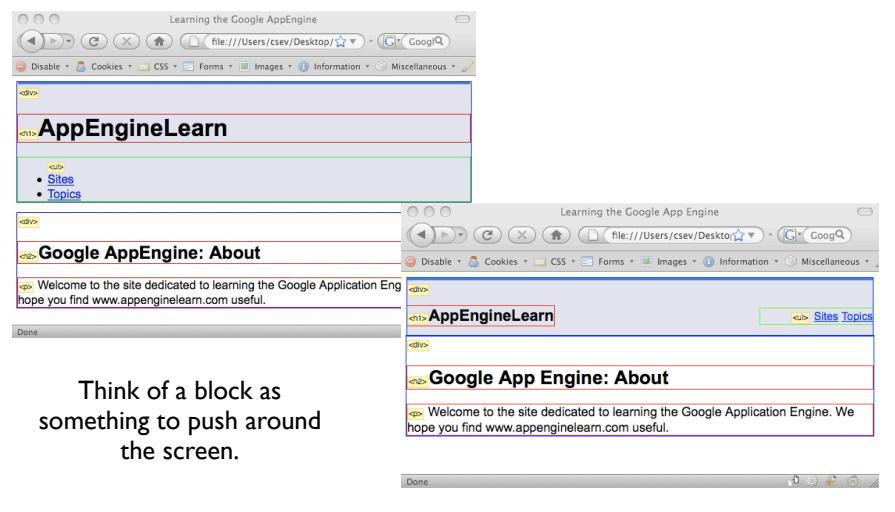
## Inline Elements

- Flowed with other text
- span, em, strong, cite, a
- Inline tags can be nested as long as they match
  - <span class="important"><cite>Stuff</cite></span>
- Block can contain inline - but inline cannot contain block

# Block Level Elements

- Starts on its own line - ends justification and starts a new block
- Can be a container for other elements
- h1 - h6, p, div, blockquote, ul, ol, form
- Blocks can contain other blocks

```
<div id="content">
  <p>One </p>
  <p>Two</p>
</div>
```



Now we will move things around, add background and borders, etc.

```
<body>
  <div id="header">
    <h1><a href="index.htm">AppEngineLearn</a></h1>
    <ul>
      <li><a href="sites.htm">Sites</a></li>
      <li><a href="topics.htm">Topics</a></li>
    </ul>
  </div>
  <div id="content">
    <h2>Google App Engine: About</h2>
    <p>
      Welcome to the site dedicated to learning the Google Application Engine.
      We hope you find www.appenginelearn.com useful.
    </p>
  </div>
</body>
```

```

body {
  font-family: arial, sans-serif;
}
#header {
  background-color: #dde;
  border-top: 3px solid #36c;
}
a{
  color: blue;
}
#header h1 a {
  text-decoration: none;
  color: black;
}

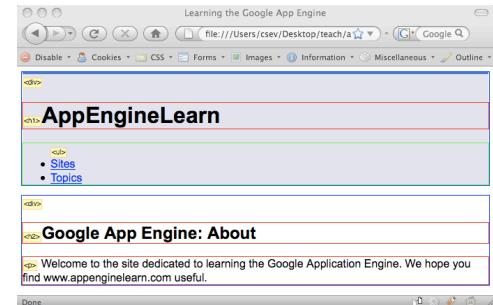
```



```

<body>
<div id="header">
<h1><a href="index.htm">AppEngineLearn</a>
<ul>
<li><a href="sites.htm">Sites</a>
<li><a href="topics.htm">Topics</a>
</ul>
</div>
<div id="content">
<h2>Google App Engine: About</h2>
<p>Welcome to the site dedicated to learning the Google Application Engine. We hope you find www.appenginelearn.com useful.</p>
</div>
</body>

```

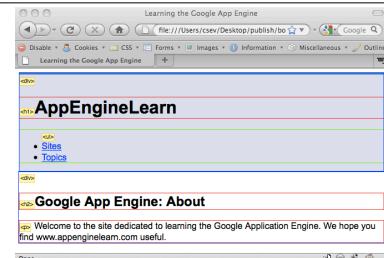


Next we will move these blocks around.

```

#header {
  background-color: #dde;
  border-top: 3px solid #36c;
  height: 100%;
  overflow:hidden;
}
#header li {
  font-size: 14px;
  display: inline;
}
#header ul {
  list-style: none;
  float:right;
  vertical-align: middle;
}
#header h1 {
  font-size: 20px;
  float: left;
  vertical-align: middle;
}

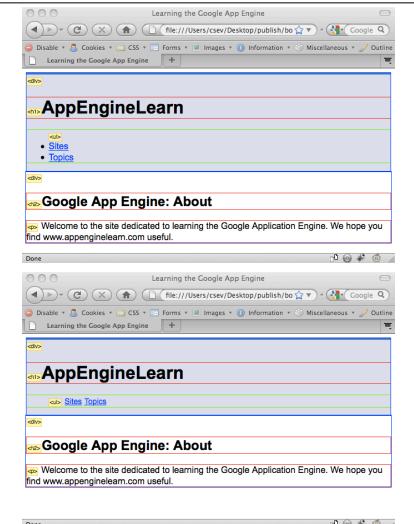
```



```

#header {
  background-color: #dde;
  border-top: 3px solid #36c;
  height: 100%;
  overflow:hidden;
}
#header li {
  font-size: 14px;
  display: inline;
}
#header ul {
  list-style: none;
  float:right;
  vertical-align: middle;
}
#header h1 {
  font-size: 20px;
  float: left;
  vertical-align: middle;
}

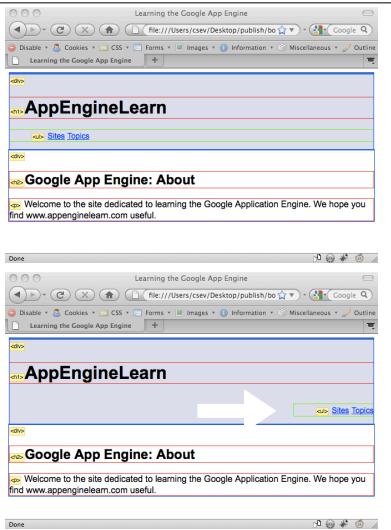
```



```

#header {
  background-color: #dde;
  border-top: 3px solid #36c;
  height: 100%;
  overflow: hidden;
}
#header li {
  font-size: 14px;
  display: inline;
}
#header ul {
  list-style: none;
  float: right;
  vertical-align: middle;
}
#header h1 {
  font-size: 20px;
  float: left;
  vertical-align: middle;
}

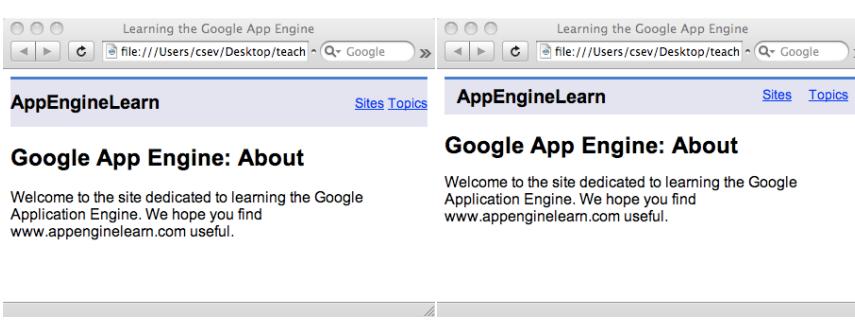
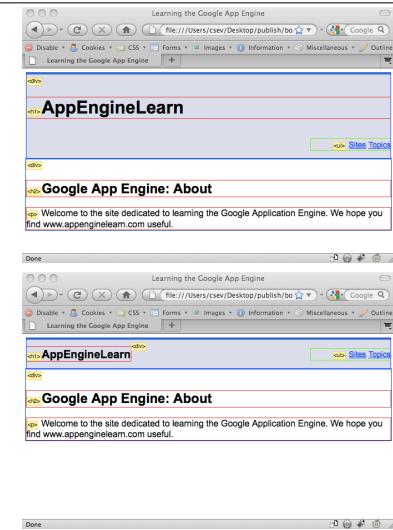
```



```

#header {
  background-color: #dde;
  border-top: 3px solid #36c;
  height: 100%;
  overflow: hidden;
}
#header li {
  font-size: 14px;
  display: inline;
}
#header ul {
  list-style: none;
  float: right;
  vertical-align: middle;
}
#header h1 {
  font-size: 20px;
  float: left;
  vertical-align: middle;
}

```



Now lets do some “tweaking” to make it more visually pleasing...

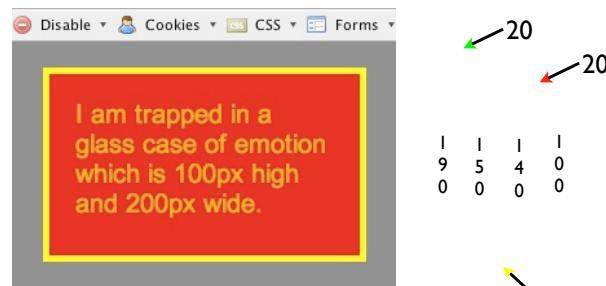
## CSS Box Model

- height and width properties size the block element
- margin properties define the space around the block element
- border properties define the borders around a block element
- padding properties define the space between the element border and the element content
- background properties allow you to control the background color of an element, set an image as the background, repeat a background image vertically or horizontally, and position an image on a page

<http://reference.sitepoint.com/css/boxmodel>

```
.trapped {
  height: 100px;
  width: 200px;
  background:red;
  padding: 20px;
  border: 5px solid yellow;
  margin: 20px;
  font-family:Arial;
  color:orange;
  font-size:20px;
}
```

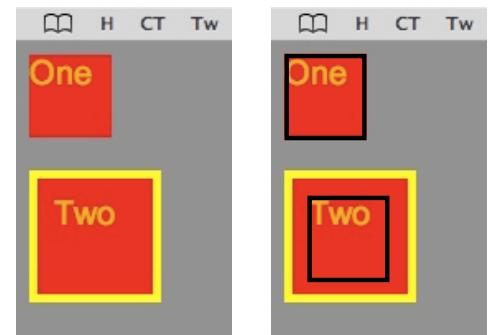
<p class="trapped">  
I am trapped in a glass case of emotion  
which is 100px high and 200px wide.  
</p>



## The Box Model

```
.trapped {
  height: 50px;
  width: 50px;
}
.trapped2 {
  height: 50px;
  width: 50px;
  padding: 10px;
  border: 5px solid yellow;
}
```

<p class="trapped">  
One</p>  
<p class="trapped2">  
Two</p>



Border, padding, and margin are additive.

```
#header {
  background-color:#dde;
  border-top: 3px solid #36c;
  height: 100%;
  overflow:hidden;
  padding: 7px;
  margin-top: 5px;
}
#header h1 {
  font-size: 20px;
  float: left;
  vertical-align: middle;
  margin: 0;
  padding: 0 0 0 .3em;
}
```

```
#header li {
  font-size: 14px;
  display: inline;
  padding: .5em;
}
#header ul {
  list-style: none;
  text-align: right;
  float:right;
  vertical-align: middle;
  margin: 0;
  padding: 0;
}
```

top, right, bottom, left

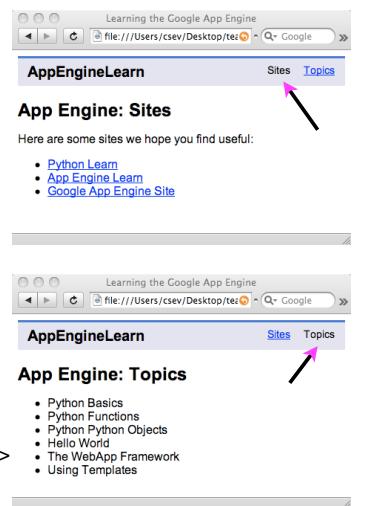


Giving Navigation Clues...

```
<div id="header">
  <h1><a href="index.htm">AppEngineLearn</a></h1>
  <ul>
    <li><a href="sites.htm" class="selected">Sites</a></li>
    <li><a href="topics.htm" >Topics</a></li>
  </ul>
```

```
<div id="header">
  <h1><a href="index.htm">AppEngineLearn</a></h1>
  <ul>
    <li><a href="sites.htm">Sites</a></li>
    <li><a href="topics.htm" class="selected">Topics</a></li>
  </ul>
</div>
```

```
#header li a.selected {  
    color: black;  
    text-decoration: none;  
}
```



## Transforming the look and feel of a page using a CSS style sheet.

Learning the Google App Engine

## AppE

## App Engine: Sites

Here are some sites we hope you find useful:

- [Python Learn](#)
- [App Engine Learn](#)
- [Google App Engine Site](#)

```
body {  
  font-family: arial, sans-serif  
}  
a, a:link {  
  color: #0000cc;  
}
```

Learning the Google App Engine

## Google App Engine: About

Welcome to the site dedicated to learning the Google Application Engine. We hope you find [www.appspotenginelearn.com](http://www.appspotenginelearn.com) useful.

# CSS Validation

- You can validate your CSS to make sure it has no syntax errors
- Browsers will generally quietly ignore bad CSS syntax
- <http://jigsaw.w3.org/css-validator>
- The validator can save you time and sanity

W3C CSS Validator results for file:///localhost/TextArea (CSS level 2.1)

Jump to: Validated CSS

W3C CSS Validator results for file:///localhost/TextArea (CSS level 2.1)

**Congratulations!** No Error Found.

This document validates as **CSS level 2.1**!

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<img alt="W3C CSS Validation Service logo" href="http://www.w3.org/Style/Validate/Result/ValidIcon.gif" style="border: 1px solid black; margin-right: 5px; vertical-align: middle;"/> CSS level 2.1
```

W3C CSS Validation Service

# Zen Garden

- A social site where CSS designers show their “stuff”
- The HTML never changes - only the CSS
- Inspiration for us designers

<http://www.csszengarden.com/>

## CSS Summary

- CSS Layout is its own art and science
- CSS Basics are well established and well supported in all modern browsers
- The box model is pretty straightforward - and allows nice design within the standards with reasonable effort levels.
- Site layout and markup is further evolving - mostly to make it increasingly possible to support desktop like experiences on the web.
- These innovations will naturally cause incompatibilities - which make things interesting and frustrating at times.